

ИГРОВОЙ СЕРВЕР: ЧТО ТАМ ВНУТРИ?

Марк Локшин
Старший программист







ПЛАН ДОКЛАДА

- Компоненты игрового сервера
- Используемый стек технологий
- Особенности обновления
- Структура игрового кластера
- Производительность



Наша студия

- 200+ сотрудников
- 15+ успешных проектов
- 100+ миллионов игроков

IT TERRITORY
GAMES MADE WITH HEART

MY.GAMES | IT TERRITORY GAMES MADE WITH HEART



IT TERRITORY

- Аллоды Онлайн
- Hawk: Freedom Squadron
- Space Justice
- World Above
- Rush Royale



КОМПОНЕНТЫ ИГРОВОГО СЕРВЕРА

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	
• Слой хранения данных	• Слой хранения данных

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP, Microsoft Dynamics AX, 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay, Pay
• Реклама и кросс-проектные связи	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1C
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	
• Поиск пути	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	
• Поиск пути	
• Подбор соперника (матчмейкинг)	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP, Microsoft Dynamics AX, 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay, Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	
• Поиск пути	
• Подбор соперника (матчмейкинг)	
• Система логирования	

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	
• Поиск пути	
• Подбор соперника (матчмейкинг)	
• Система логирования	
• Инструментарий управления игрой	• «Админка» сайта

ИЗ ЧЕГО СОСТОИТ ИГРОВОЙ СЕРВЕР



Компонент	Внеигровые приложения
• Авторизация	
• Игровая механика (метаигра + верификация действий)	SAP Microsoft Dynamics AX 1С
• Слой хранения данных	• Слой хранения данных
• Платежи	G Pay Pay
• Реклама и кросс-проектные связи	
• Почта	
• Чат	
• Поиск пути	
• Подбор соперника (матчмейкинг)	
• Система логирования	
• Инструментарий управления игрой	• «Админка» сайта
• Продуктовая аналитика	



ИСПОЛЬЗУЕМЫЙ СТЕК ТЕХНОЛОГИЙ



ЧТО БЫЛО ИЗНАЧАЛЬНО?

- 2007 год
- Pure Java SE
- Проприетарные протоколы
- Проприетарное API
- Библиотеки Hibernate, GWT, Trove collections, Protobuf …



ИНТЕРЕСНО, НО НЕ ЭФФЕКТИВНО

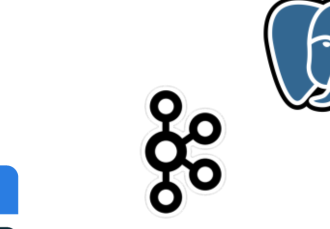


- Много интересных вещей кодогенерация, правка байт-кода на лету
- Отсутствие документации
- Отсутствие общих руководств
- ~~Stack Overflow Driven Development~~
- Нужен функционал – запили.
- Сложность поддержки





ЧТО МЫ ИСПОЛЬЗУЕМ?









- PostgreSQL
- Photon Cloud
- Kafka
- Hazelcast
- Vert.X
- Prometheus + Grafana



- Игровая механика
- Верификация действий



PHOTON CLOUD



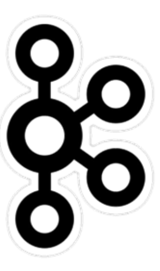
- «Комнаты» для игры
- Дата-центры в X точках на всех континентах (пингвинов обидели)
- Динамическое выделение железа
- Заточен под игры, но не только (например – текст/звук/видеочат)



- Слой хранения данных
- Система логирования



APACHE KAFKA



Распределенный брокер сообщений

- Горизонтальное масштабирование
- Хранение сообщений
- Репликация



- Слой хранения данных
- Игровая механика
- Метаигра
- Чат



HAZELCAST

In memory Data Grid



- Интегрирован в Vert.X
- Работа в кластере
- Балансировка
- Горизонтальное масштабирование
- Добавление/отказ узла
- Распределенные структуры данных
- Запросы



- Подбор соперника
- Игровая механика
- Метаигра
- Верификация действий
- Чат
- Почта
- Платежи



VERT.X

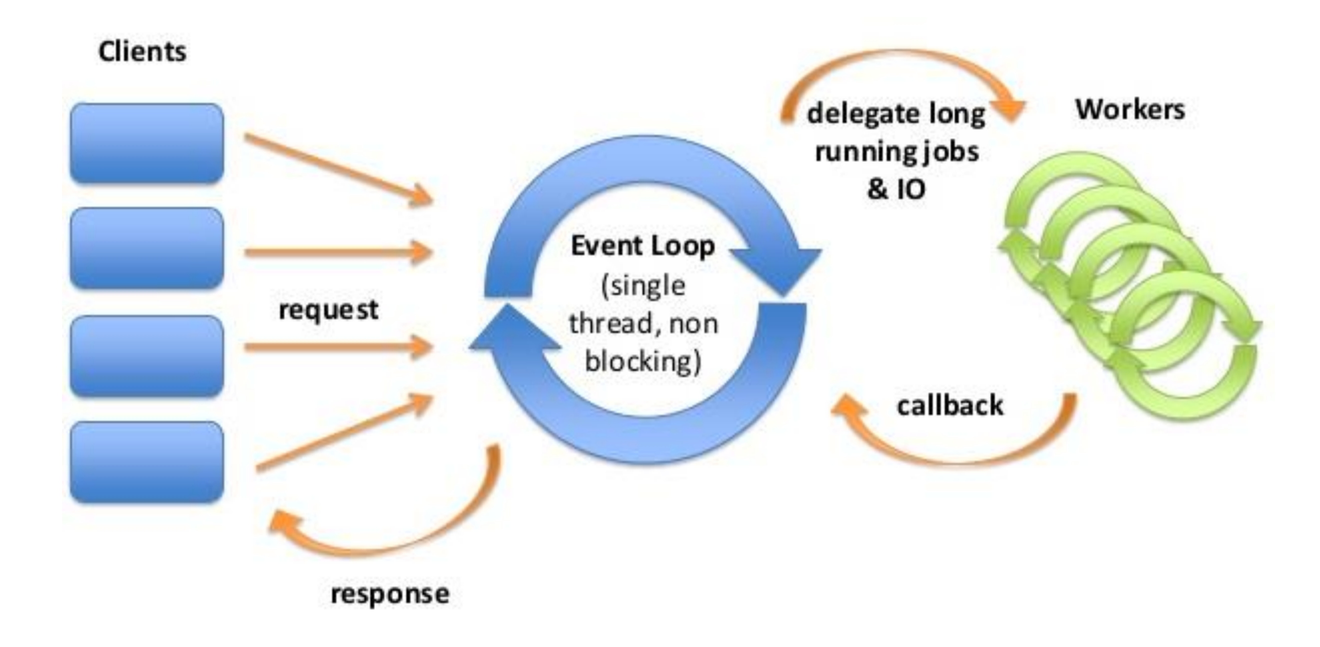


- Свободно-распространяемый фреймворк от Eclipse для построения реактивных распределенных событийно-ориентированных приложений, работающих на JVM
- Verticle – однопоточный сервис
- Распределенная шина сообщений
- Асинхронность
- Параллелизм
- Мультиязычность



ШАБЛОН РЕАКТОР







OPERATION EXECUTOR



- Объявляем сущности, с которыми работаем (можно потребовать эксклюзивного доступа через блокировки)
- Берем необходимые блокировки
- Достаем из Hazelcast указанные сущности
- Исполняем код операции
- Отправляем в Hazelcast измененные сущности
- Снимаем блокировки
- Исполняем callback на завершение операции



DEADLOCK ПРИ БЛОКИРОВКЕ

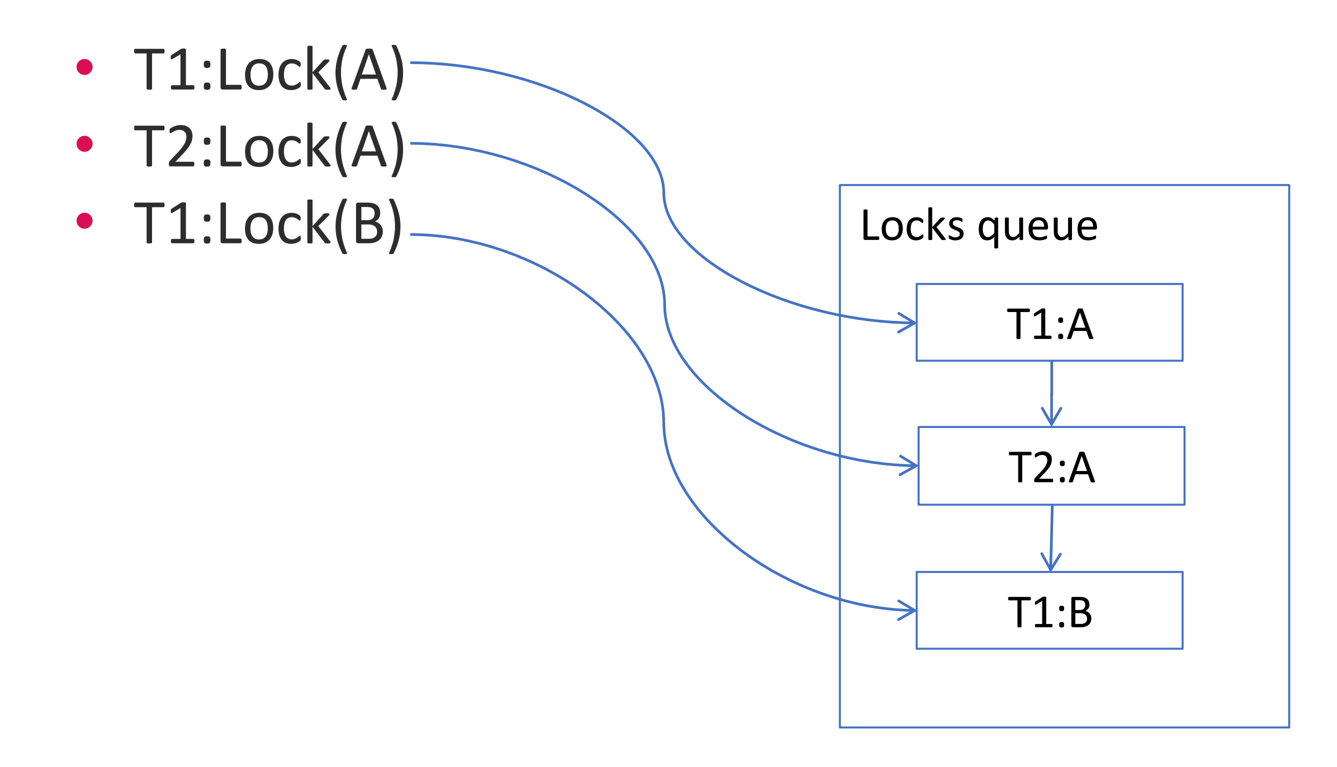


- Операция 1 берет блокировку на ресурс A
- Операция 2 пытается взять блокировку на ресурс A и ждет
- Операция 1 пытается взять блокировку на ресурс B
- … и внезапно deadlock



ГДЕ-ТО ВНУТРИ VERT.X







DEADLOCK ПРИ БЛОКИРОВКЕ



- https://groups.google.com/g/vertx/c/-uvLMuubpz8/m/OB2JzdDdAwAJ
- https://github.com/vert-x3/vertx-hazelcast/issues/41







ПОЯСНЕНИЯ ОТ АВТОРА



The original reason executeBlocking defaults to ordered=true (it's more general than just hazelcast usage) is something like this:

Imagine you have a web application and request #1 comes in - this requests inserts some data into a database (e.g. add to shopping basket)
Immediately after this requests #2 comes in - this selects the same data from the database (e.g. view shopping basket)



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность
- Все очень плохо…



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность
- Все очень плохо…
- БД??? Убираем работу с базой



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность
- Все очень плохо…
- БД??? Убираем работу с базой
- Все очень плохо…



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность
- Все очень плохо…
- БД??? Убираем работу с базой
- Все очень плохо…
- Продолжаем тестировать в разных конфигурациях производительность



НАГРУЗОЧНОЕ ТЕСТИРОВАНИЕ VERT.X



- Написали сервер, который исполняет операцию
- Написали клиент, который шлет операцию на сервер
- Тестируем в разных конфигурациях производительность
- Все очень плохо…
- БД??? Убираем работу с базой
- Все очень плохо…
- Продолжаем тестировать в разных конфигурациях производительность
- Все очень плохо… но иногда хорошо…

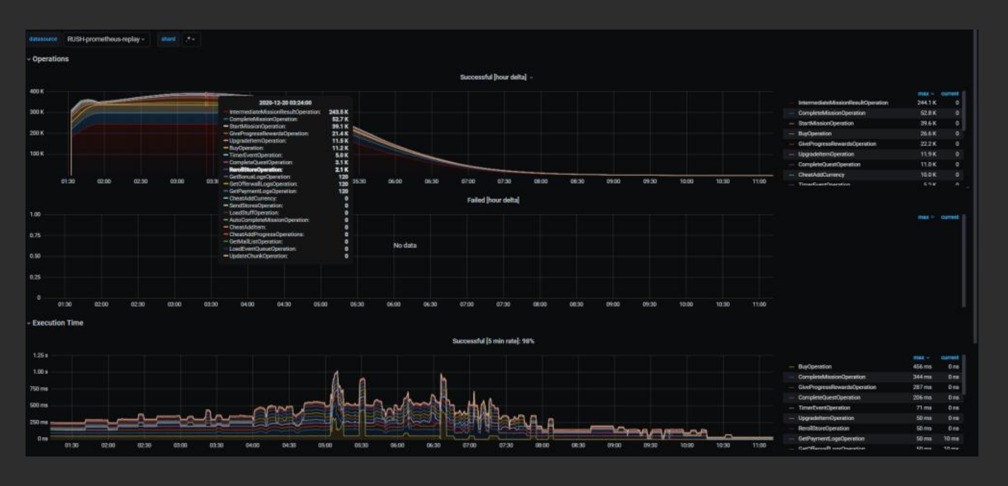


PROMETHEUS + GRAFANA











«ГОНКА» В VERT.X ПРИ СОЗДАНИИ КАНАЛА



- Внутри Vert.X при создании канала код брал блокировки
- Когда одновременно создавалось много каналов – код работал очень долго
- Мы создавали по каналу на пользователя – при большом количестве пользователей создание каналов шло очень медленно
- Переделали схему создания каналов на каналы по сообщениям
- Сделали фикс для создания каналов в Vert.X
- Сделали пул-реквест для работы Vert.X + Prometheus



ОСОБЕННОСТИ ОБНОВЛЕНИЯ



ОБНОВЛЕНИЕ ИГРЫ





Бизнес-требования

- Время простоя → min
- Цена разработки → min
- Выручка → max



ПРОЦЕСС ОБНОВЛЕНИЯ





- Задержка доступности обновления
- Шарды с разными версиями
- Единая БД
- «Мягкое» обновление
- «Жесткое» обновление



ПЕРЕХОД ИГРОКОВ НА НОВУЮ ВЕРСИЮ

8 12:00
02/09 00:00
02/09 12:00
02/10 00:00
02/10 12:00



СТРУКТУРА ИГРОВОГО КЛАСТЕРА

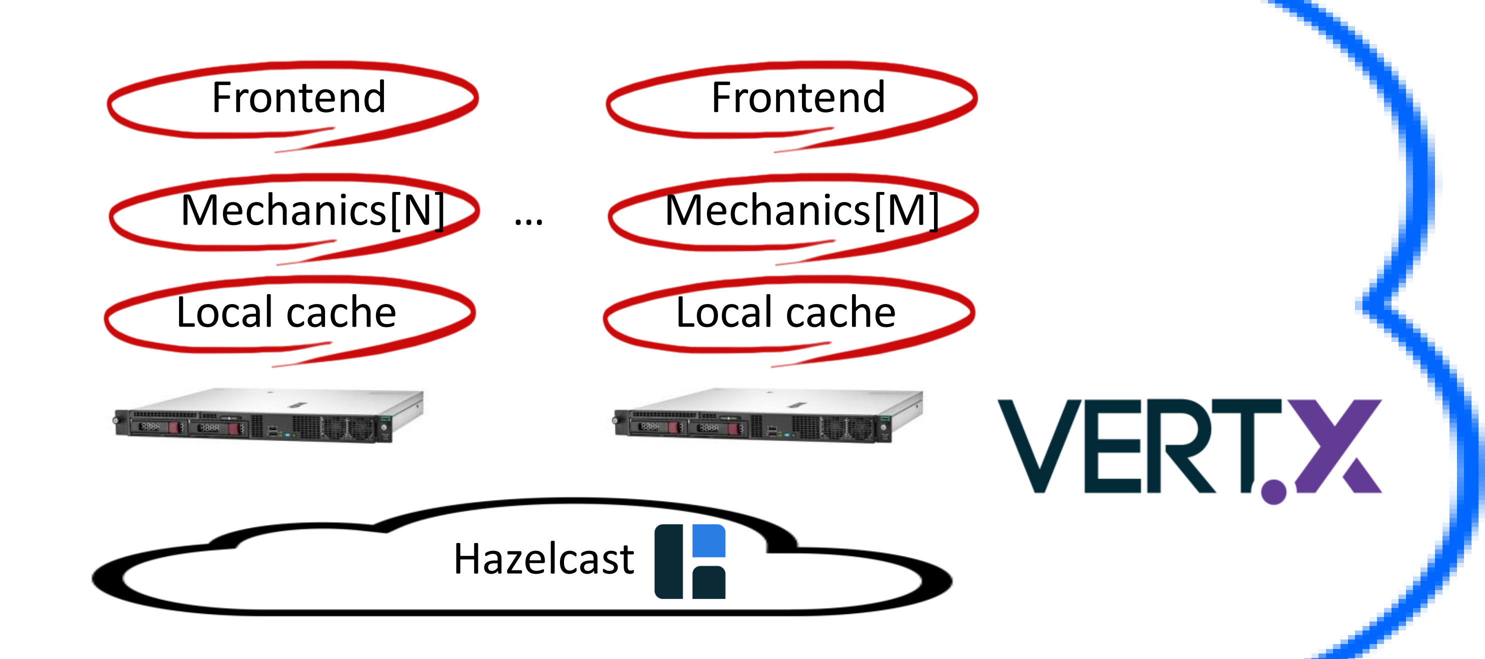


GAME SHARD



Shard

=





ИНФРАСТРУКТУРА ПРОЕКТА

Account Server

VERT.X

Shard 1

Shard 2

Shard 3

Hazelcast

Kafka

hadoop

ETL

Gametool Web Server

Account

Lobby

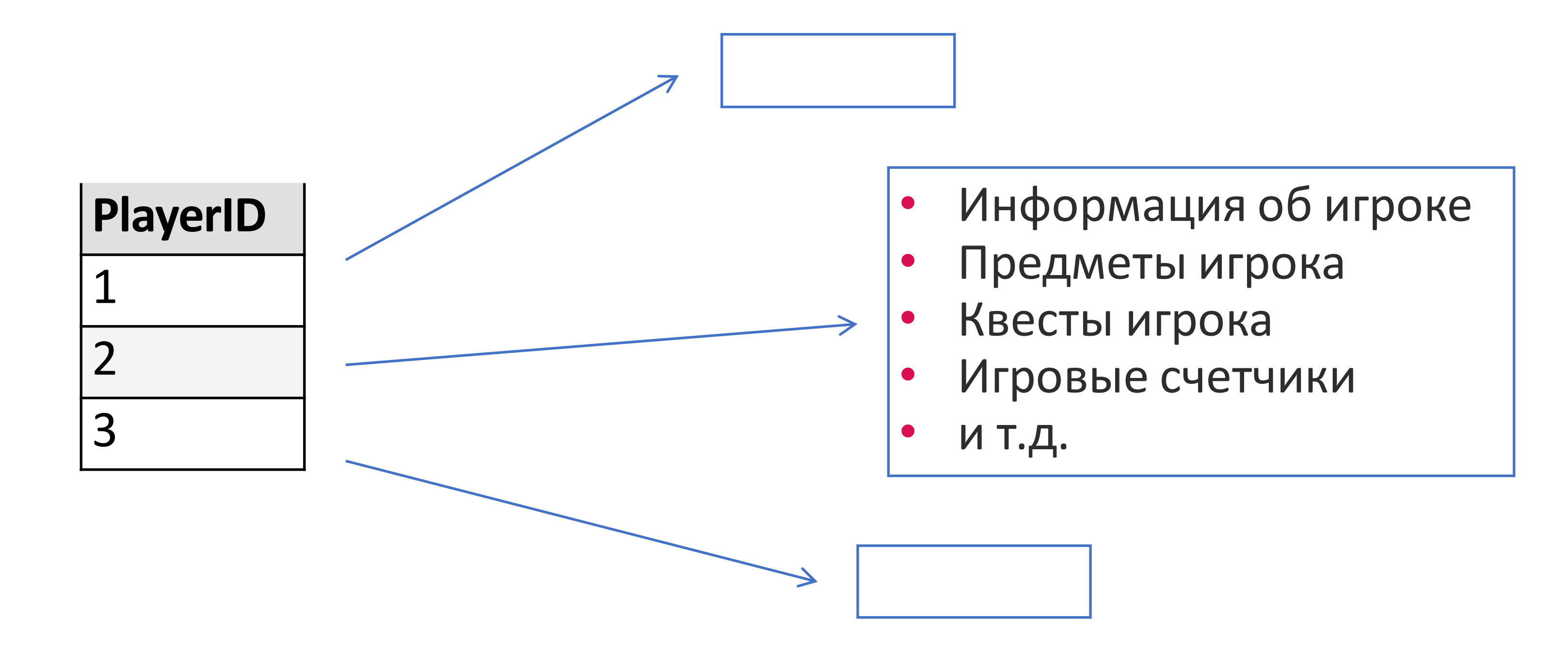
Gametool



СТРУКТУРЫ В HAZLECAST

Внутри игрового шарда (описание игрока)







СТРУКТУРЫ В HAZLECAST

Общий для шардов (например, турнир)

PlayerID
1
2
3

- История боев
- История боев
- История боев

№ корзины
1
2
3

- Игроки
- Очки
- Попытки

PlayerID	№ корзины
1	1
2	1
3	2

Счетчик игроков	3

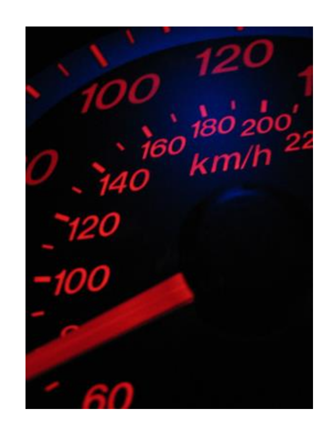


ПРОИЗВОДИТЕЛЬНОСТЬ



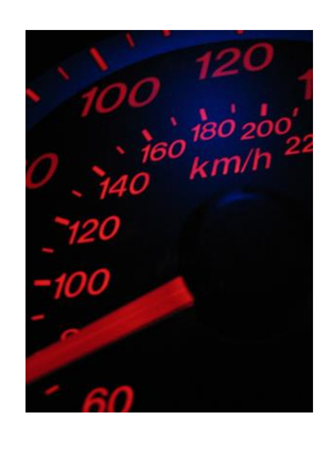
50K АКТИВНЫХ ПОЛЬЗОВАТЕЛЕЙ

- Game Servers: 2x8 2.5GHz Xeon + 32Gb RAM x3
- GameDB Server: 1x8 2.5GHz Xeon + 256Gb RAM
- Account Server: 1x8 2.5GHz Xeon + 4Gb RAM





50K АКТИВНЫХ ПОЛЬЗОВАТЕЛЕЙ



- Game Servers: 2x8 2.5GHz Xeon + 32Gb RAM x3
- GameDB Server: 1x8 2.5GHz Xeon + 256Gb RAM
- Account Server: 1x8 2.5GHz Xeon + 4Gb RAM
- AccountDB Server
- Kafka
- ETL
- GametoolDB Server
- Gametool WEB Server
- Nginx



ВАРИАНТЫ МАСШТАБИРОВАНИЯ



- CPU – горизонтальное добавление железа в кластер
- БД – vanilla PostgreSQL → нет шардирования → лучше железо
- Запись в IMDG, синхронизация с PostgreSQL отложенная
- Разные шарды



ВЫВОДЫ



- Стек технологий хорошо масштабируется и реально работает
- Ошибки встречаются
- Универсальность хорошо, но не всегда



МАРК ЛОКШИН

Старший программист

m.lokshin@corp.mail.ru

≡ × >

THANK YOU!